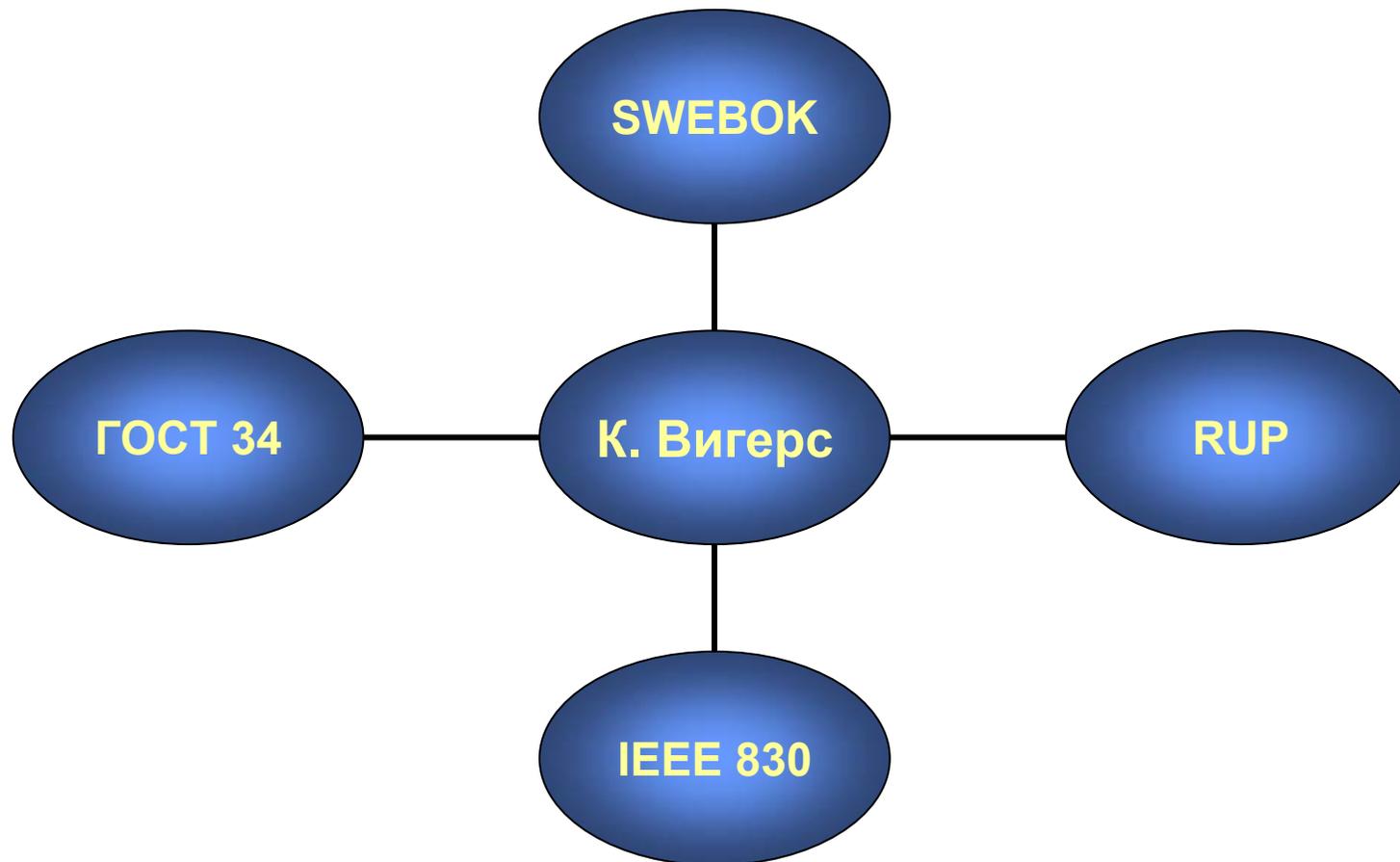


Классификация требований к программному обеспечению и ее представление в стандартах и методологиях.

Булуй Ю. И.
Независимый консультант
Yury.Buluy@gmail.com
и
СМ-Консалт
<http://www.cmcons.com/>

Содержание



Что такое «требование» ?

- Условие или возможность требуемая пользователем для решения задач или достижения целей.
- Условие или возможность, которые должны удовлетворяться системой/компонентом системы или которыми система/компонент системы должна обладать для обеспечения условий контракта, стандартов, спецификаций или др. регулирующих документов.
- Документальная репрезентация условий или возможностей, перечисленных в пред. пунктах.

IEEE Std 610.12, "IEEE Standard Glossary of Software Engineering Terminology"

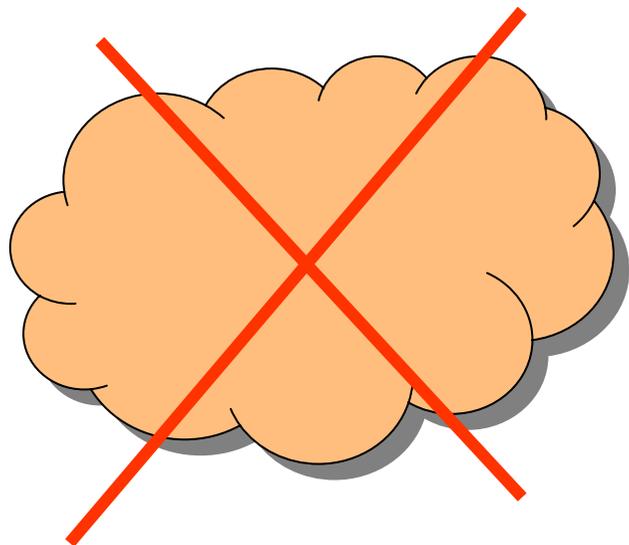
«Требования это спецификация того что должно быть получено. Требования описывают поведение системы или атрибуты и свойства системы. Требования могут являться и ограничениями на процесс разработки системы»

Ian Sommerville & Pete Sawyer

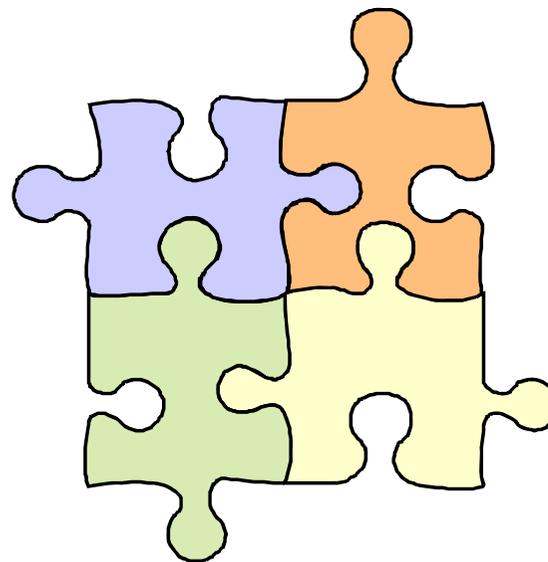
«Условие или возможность которой должна удовлетворять система» (A requirement is defined as "a condition or capability to which a system must conform").

Rational Unified Process, Version 2003.06.13

Дискретность требований



Бесформенный набор предложений



Отдельные (дискретные) утверждения, составляющие целостную картину

Дискретность вытекает в т.ч. и из «характеристик хороших требований» и свойства трассируемости требований.

Предпосылки

- При большом количестве требований в конкретном проекте **навигация** и **понимание сути проекта**, без организации информации каким-либо образом, становится затруднительной.
- Требование может описывать те или иные **характеристики или свойства программного продукта**.
- Требование само по себе может иметь **набор свойств (атрибутов)**.
- Требования могут иметь **сложную структуру отношений и связей** (например иерархическую).

Знание начинается с ... классификации!

- **Классификация** – система или способ группировки субъектов наблюдения или исследования в соответствии с их общими признаками
(ru.wikipedia.org/wiki/Классификация)
- Природа требований позволяет их классифицировать, остается только определить классификационные признаки.
- Наличие классификации позволяет использовать общие шаблоны (паттерны?) для работы с требованиями (общие наборы атрибутов, правила отношений и связей, ...)

Требования в SWEBOOK



Классификация требований в SWEBOOK

- **Требования к продукту и процессу** – параметры относящиеся к продукту или процессу его создания.
- **Функциональные и нефункциональные требования:**
 - Функциональные описывают функции которые выполняет ПО
 - Нефункциональные требования накладывают определенные ограничения
- **Независимые свойства** – требования которые не могут быть адресованы к одному из компонентов системы, а проявляются при взаимодействии.
- **Системные или программные требования.** Относятся к системе в целом или к программной составляющей.

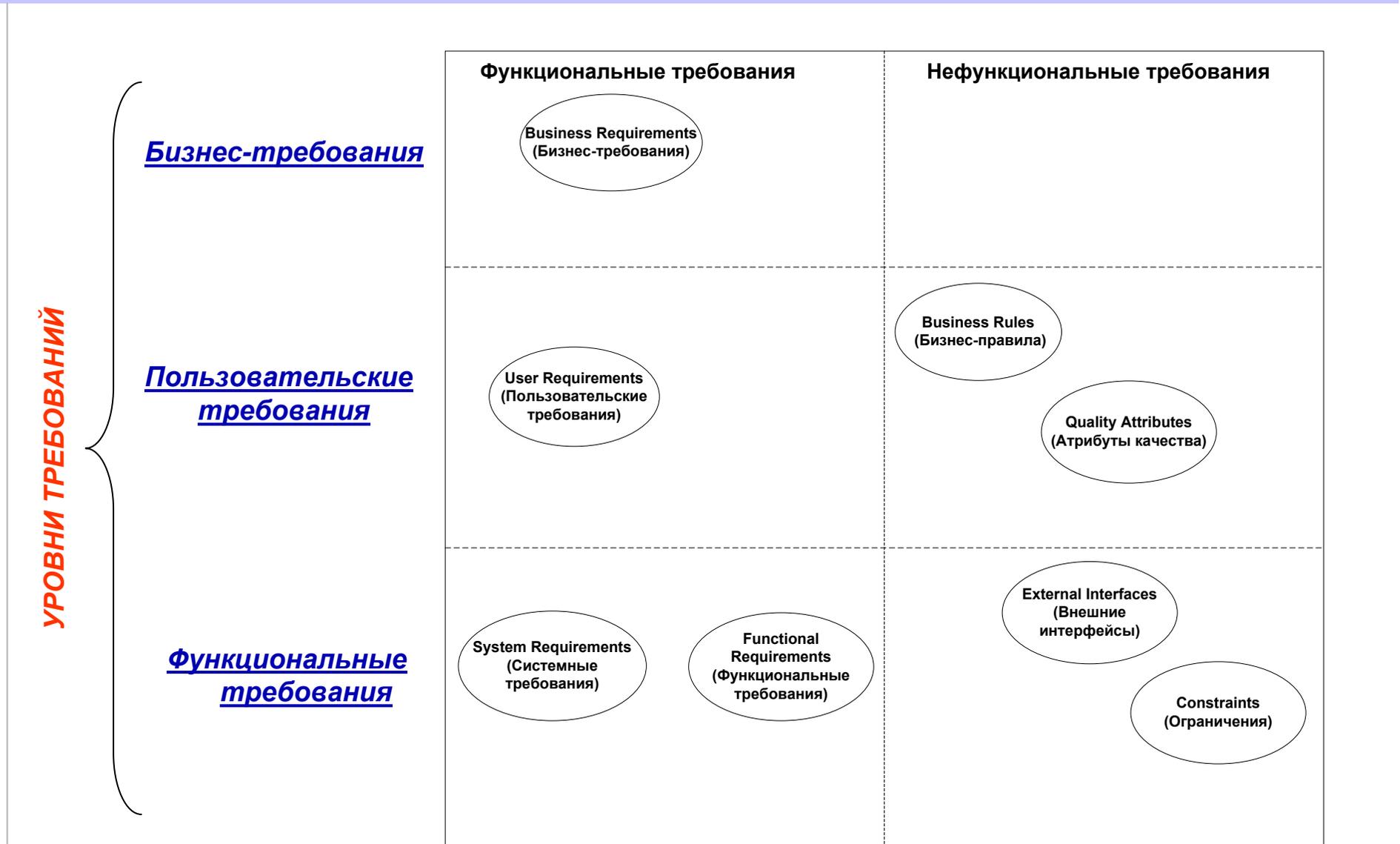
«Dimensions»

- Отношение требования к функциональным или нефункциональным;
- Является ли требование производным от более высокоуровневых требований, является ли «независимым свойством» или было непосредственно получено от заинтересованных лиц;
- Требования к продукту или к процессу;
- Приоритет требований;
- По области действия (глобальные, на компонент, ...);
- По степени стабильности;

Выводы по SWEBOOK

- SWEBOOK не описывает в «чистом виде» подходы к классификации требований как таковой, а в большей степени описывает возможности по группировке требований в соответствии с их характеристиками
- SWEBOOK отмечает, что имеется явное пересечение между описанными в нем «dimensions» и атрибутами (свойствами) требований.
 - Многие «критерии» предлагаемой группировки требований по сути – группировка по значению атрибута.

Классификация требований по К. Вигерсу



УРОВНИ ТРЕБОВАНИЙ

Уровни требований

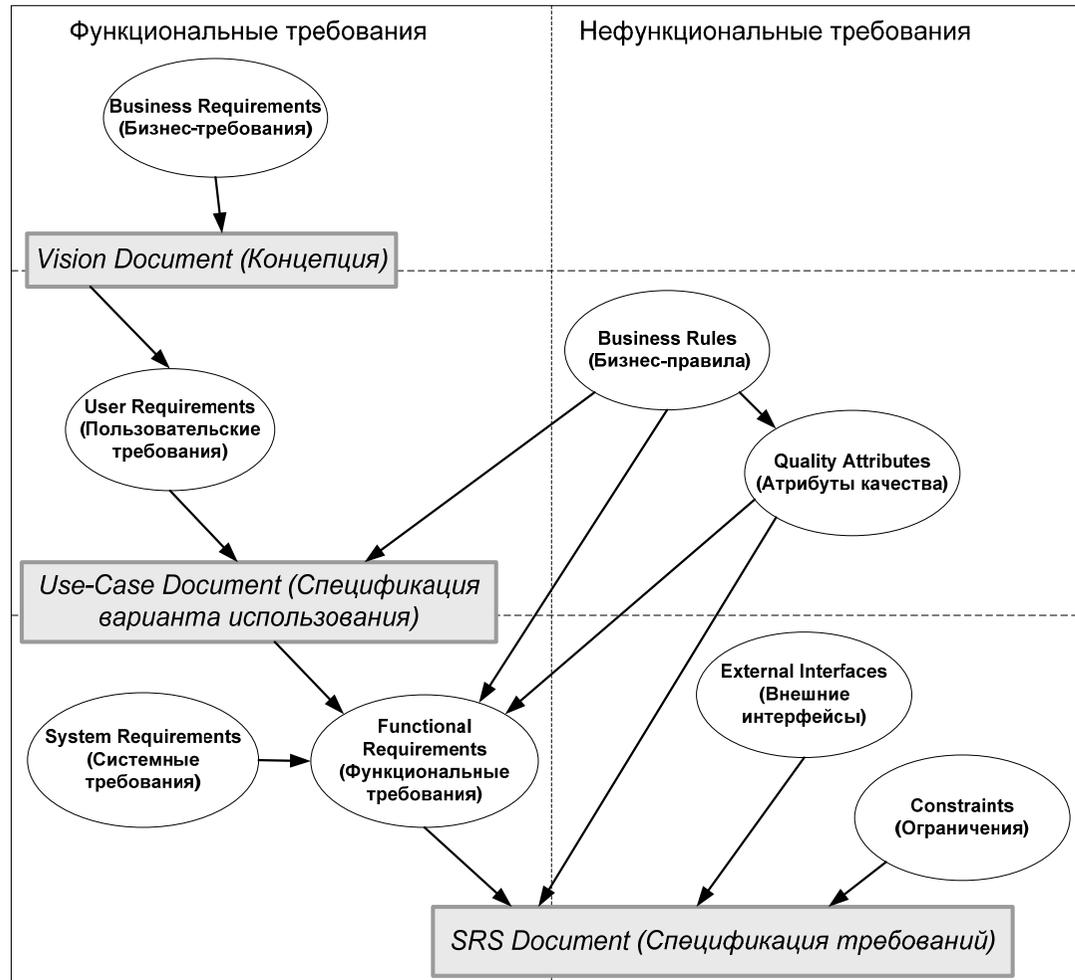
- **Бизнес-требования (Business Requirements)** определяют высокоуровневые цели организации или клиента (потребителя) – заказчика разрабатываемого ПО. (**ПОЧЕМУ?**)
- **Пользовательские требования (User Requirements)** описывают цели/задачи пользователей системы, которые должны выполняться пользователями при помощи создаваемого ПО. (**КТО и ЧТО?**)
- **Функциональные требования (Functional requirements)** определяют функциональность ПО, которая должна быть создана разработчиками для предоставления возможности выполнения пользователями своих обязанностей в рамках бизнес-требований. (**ЧТО?**)

Другие типы требований

- **Нефункциональные требования** включают эксплуатационные характеристики и описание *атрибутов качества*.
 - **Атрибуты качества (Quality Attributes)** – описывают дополнительные характеристики продукта в различных измерениях, важных для пользователя или разработчиков. Эти характеристики включают практичность, портабельность, целостность, эффективность, устойчивость.
 - **Ограничение** – это формулировка условия, которое модифицирует требование или набор требований сужая выбор возможных решений.
 - **Бизнес-правила (business rules)** включают корпоративную политику, законодательные акты, промышленные стандарты, учетную практику и алгоритмы вычислений.
- **Системные требования (system requirements)** описывают высокоуровневые требования для ПО, которое содержит много подсистем (IEEE Std 1233-1998).

Отображение на документы

- Vision
- Use Case Document
- SRS



Классификация Вигерса (примечания)

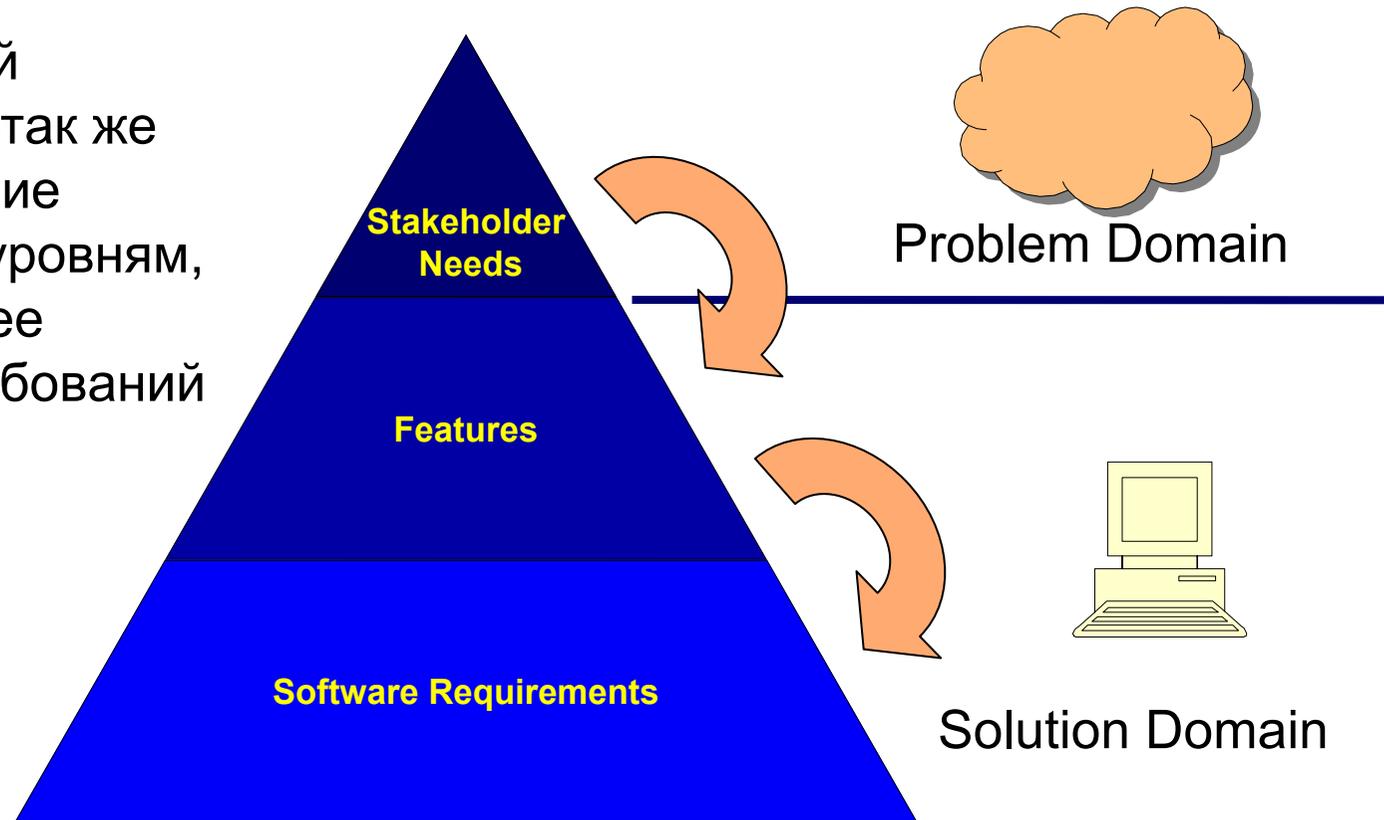
- Вигерс не употребляет термин «классификация требований», а говорит о «способе представления» и «модели»
- Такая модель, по его мнению, не является исчерпывающей, но является хорошим способом организации представления и группировки требований.

Требования в RUP

- Методология **RUP** выделяет отдельную дисциплину Requirements, которая объединяет работы, роли и артефакты (суть документы, модели, и т.п.), связанные с созданием требований к ПО и управления ими.
- RUP – процессный фреймворк. Имеет смысл рассматривать рекомендованный RUP подход к работе с требованиями.
- В основе подхода RUP к работе с требованиями лежат работы многих известных специалистов. Д. Лефингвелл представил обобщенную классификационную схему – структуру требований, которая отражает рекомендованный RUP подход.

Схема Лефингвелла

В основе данной классификации так же лежит разделение требований по уровням, соответствующее назначению требований



Уровни требований

- **Потребность (need)** – отражение проблемы бизнеса, персоналии, или процесса, которое должно быть соотнесено с использованием или приобретением новой системы
- **Характеристика продукта (feature, фича)** – множество логически связанных функциональных требований, которые обеспечивают возможности пользователя и удовлетворяют бизнес-целям.
 - Рассматривается как с т.з. маркетинга (a bullet item in the product description), или как «высокоуровневое выражение поведения системы» [**Leffingwell & Widrig,2003**]
 - features могут быть как относящимся к функциональным, так и к нефункциональным требованиям. И могут изменяться от версии к версии продукта.

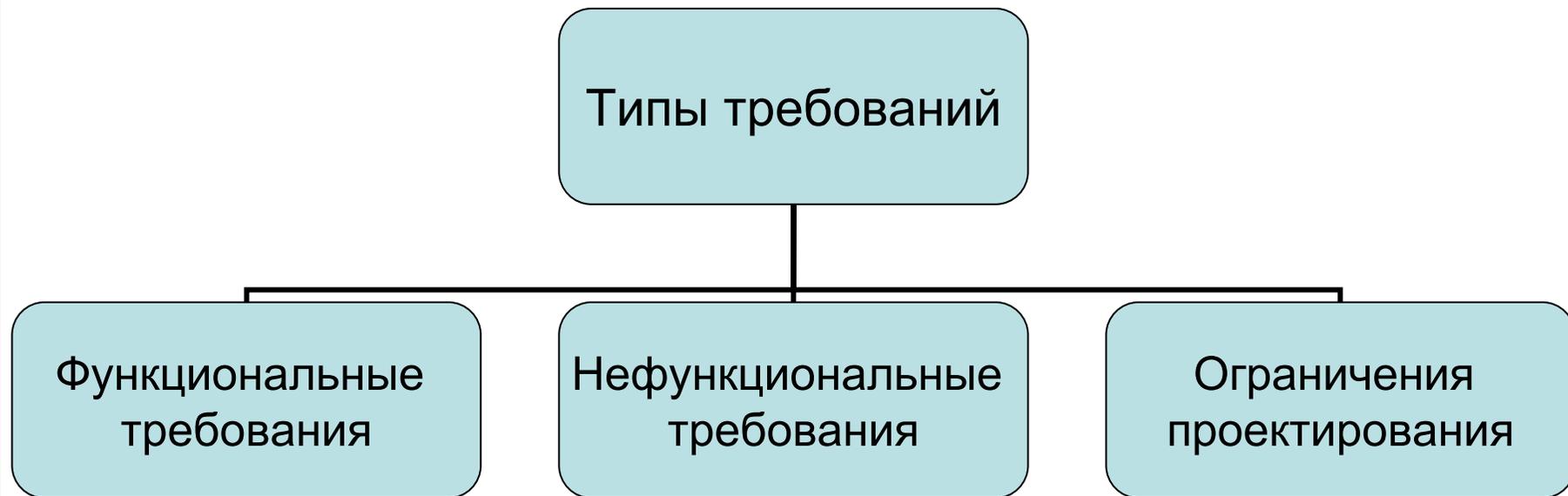
Use cases

- «Use case описывает последовательность действий выполняемых системой для получения видимого и значимого результата для конкретного действующего лица».
- Рекомендуемый RUP подход отводит значимое место UC, отождествляя их с функциональными требованиями.
- «Use cases are just one way to express software requirements». [Leffingwell & Widrig, 2003]
- «The use-case model is an effective way of expressing detailed functional "software requirements"» [RUP, 2003]

Классификация требований в RUP

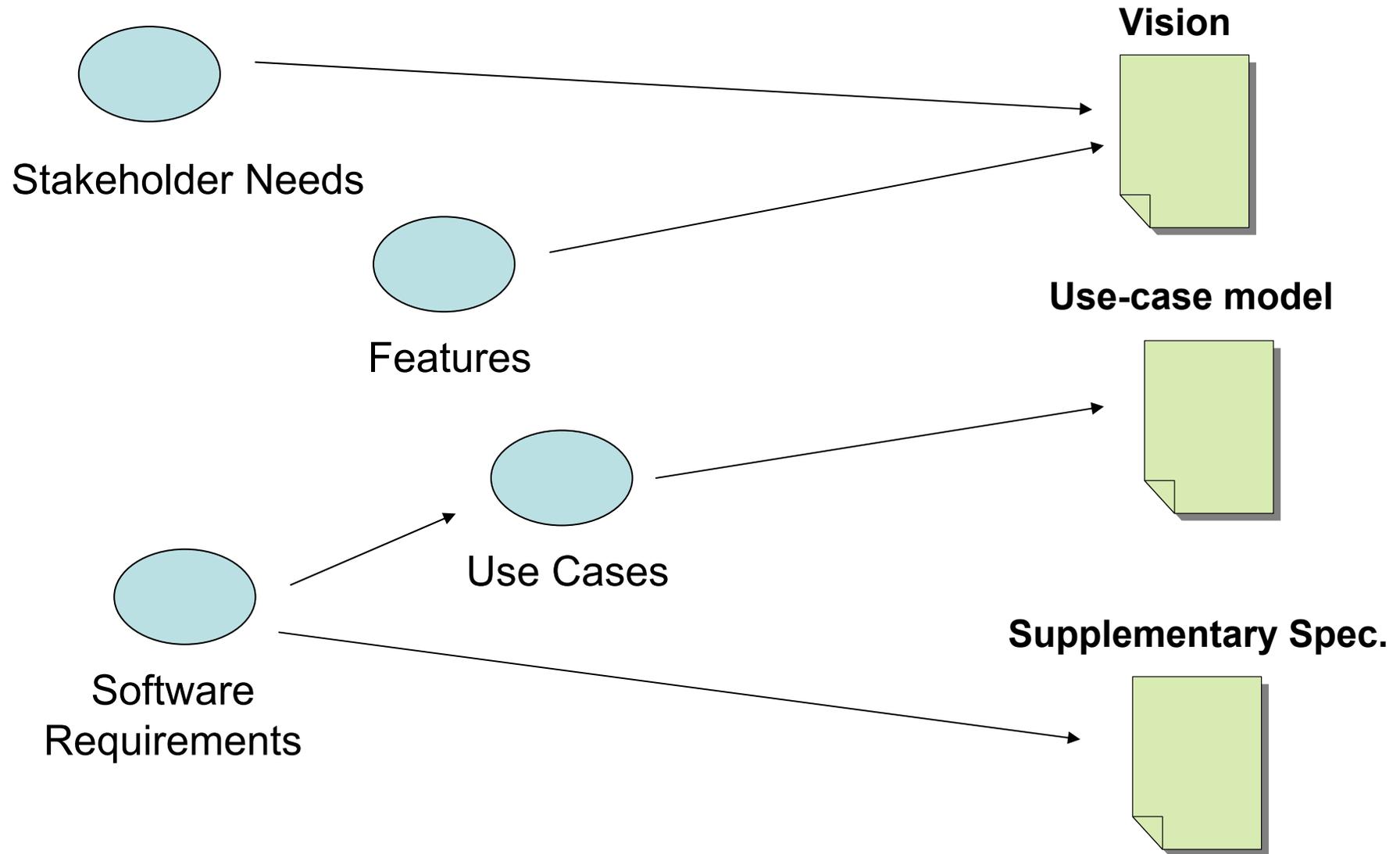
- Понятие «требование» определяется как «**условие или возможность которому должна удовлетворять система**».
- Требования могут быть категоризированы по модели **FURPS+**, которая в своем названии описывает основные категории требований с подкатегориями:
 - **F**unctionality (Функциональность)
 - **U**sability (Удобство использования)
 - **R**eliability (Надежность)
 - **P**erformance (Производительность)
 - **S**upportability (Поддержка)
- "+" в FURPS+ говорит о включении дополнительных категорий:
 - design constraints (ограничения проектирования)
 - implementation requirements (требования к реализации)
 - interface requirements (требования к интерфейсам)
 - physical requirements (требования к физическим характеристикам).
- **Функциональные требования** описывают действия которые система должна выполнять, без учета физических ограничений

Детализация Software Requirements (Леффингвелл)

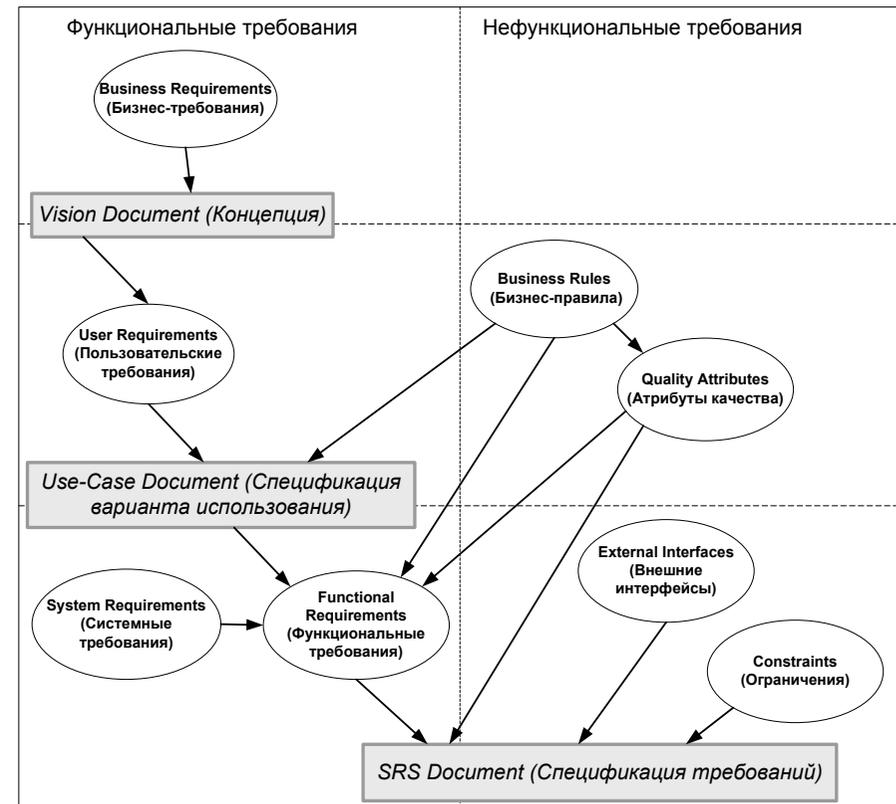
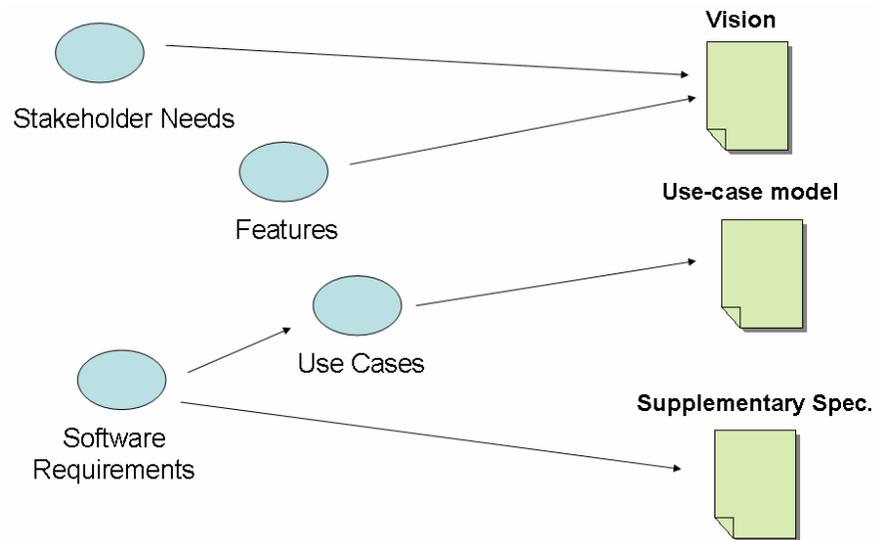


- **Функциональные требования*** выражают поведение системы – входы, выходы и функции, которые система предоставляет пользователю.
- **Нефункциональные требования** (Usability, Reliability, Performance, Supportability)
- **Ограничения проектирования** – ограничения, относящиеся к дизайну или процессу создания системы, которые не влияют на внешнее поведение, но должны быть учтены для соответствия различного вида обязательствам

Классификация и артефакты



Отображение на документы (RUP и Вигерс)



Классификация Вигерса и RUP

- RUP отождествляет понятие «требование» **только с Software Requirements.**
- **Бизнес-требования** и **Stakeholder/User Needs** – достаточно близкие понятия, относящиеся к problem domain.
- В RUP **Features** существуют как самостоятельная классификационная единица.
- Вигерс выделяет **Пользовательские требования**, отмечая, что они могут быть выражены вариантами использования (use cases).
- В RUP Software Requirements имеют более широкий «охват», включая как функциональные, так и нефункциональные требования.
- В обоих подходах может быть создано по 3 основных документа (артефакта) на которые отображаются классификации требований.

Представление требований в IEEE 830

- Стандарт IEEE 830-1998 IEEE Recommended Practice for Software Requirements Specifications описывает рекомендуемые способы спецификации требований к программному обеспечению.
- Результатом процесса спецификации требований является однозначно интерпретируемый и целостный документ-спецификация требований.
- Стандарт на документальное оформление требований, задает определенные группировки одинаковых по своей сути требований (по их назначению), фиксируя классификацию требований.

Основные элементы SRS

- Стандарт определяет основные элементы, которые должны быть отражены в Software Requirements Specifications (SRS):
 - **Функциональность**. Ответ на вопрос «ЧТО должно делать ПО?»
 - **Внешние интерфейсы**. Как ПО должно взаимодействовать с людьми, другим ПО или аппаратурой.
 - **Производительность**. С какой скоростью должны производиться те или иные операции, каково время отклика и т.п.
 - **Атрибуты**. Определяют точность, переносимость, модифицируемость, эксплуатационные качества и т.п.
 - **Ограничения проектирования**. Использование определенных языков программирования, баз данных, форматов обмена информацией и т.п.

Общая структура SRS

Table of Contents

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, acronyms, and abbreviations

1.4 References

1.5 Overview

2. Overall description

2.1 Product perspective

2.2 Product functions

2.3 User characteristics

2.4 Constraints

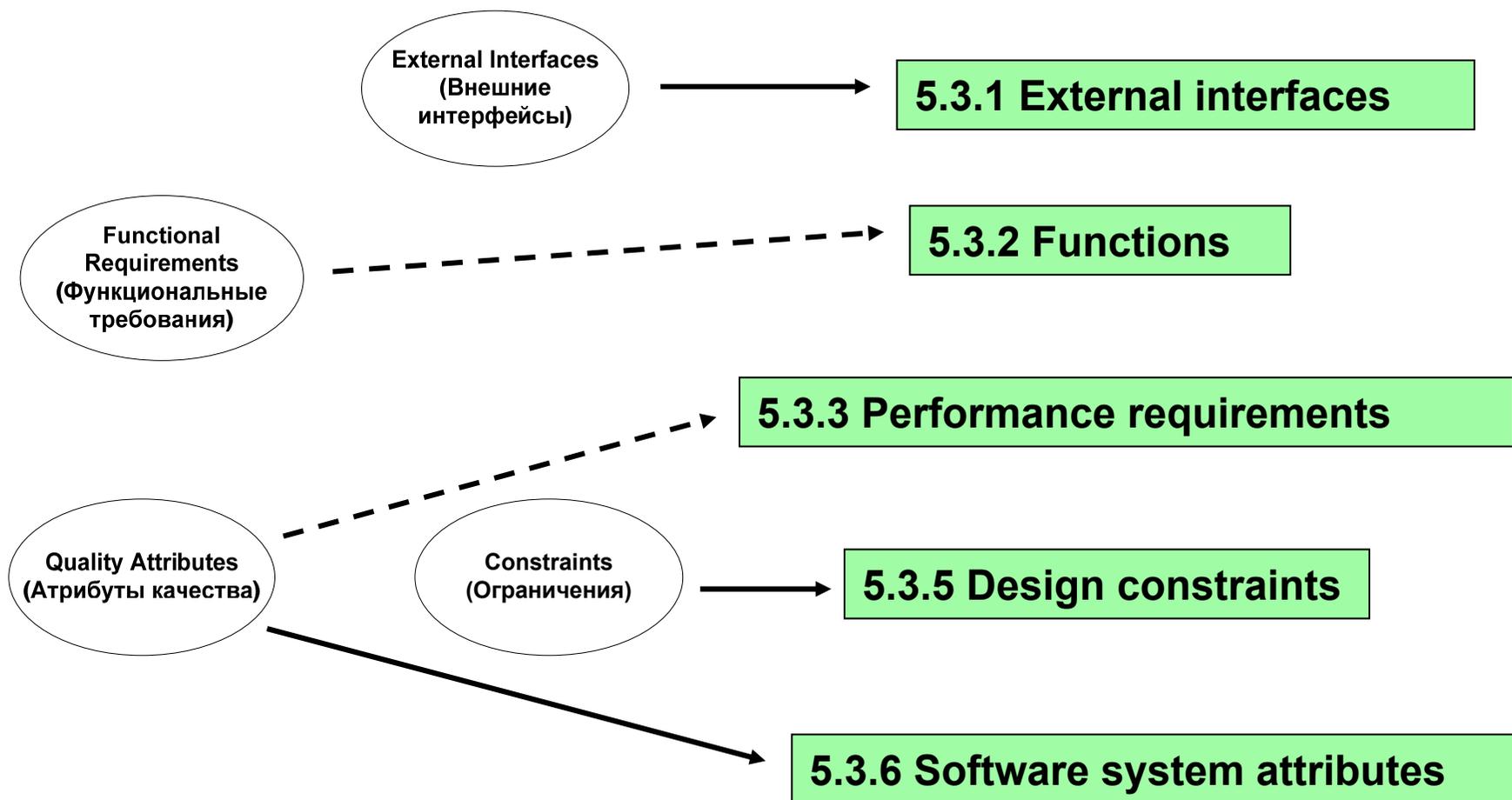
2.5 Assumptions and dependencies

3. Specific requirements (See 5.3.1 through 5.3.8 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.)

Appendixes

Index

Соответствие типов требований (Виггерс) и SRS IEEE 830



Представление требований в ГОСТ

- ГОСТ 34.602 определяет состав и правила оформления документа «Техническое задание на создание (развитие или модернизацию) системы» (см. Автоматизированная система, в ГОСТ 34.003).
- Документ Техническое задание устанавливается как основной документ, определяющий «требования и порядок создания автоматизированной системы».
- Предполагая, что на основании этого документа будет производиться разработка системы и ее приемка.

Структура документа

- ГОСТ 34.602 предлагает следующую структуру разделов документа Техническое задание:
 - общие сведения;
 - назначение и цели создания (развития) системы;
 - характеристика объектов автоматизации;
 - требования к системе;
 - состав и содержание работ по созданию системы;
 - порядок контроля и приемки системы;
 - требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;
 - требования к документированию;
 - источники разработки.
- Подраздел «Цели создания системы» и раздел «Требования к системе» имеют непосредственное отношение к отражению классификации требований.

Соотнесение типов требований (Вигерс) и ГОСТ (34.602, ...)



Описание соответствий

<p>Бизнес- требования</p>	<p>2.4.2. В подразделе «Цели создания системы» приводят наименования и требуемые значения технических, технологических, производственно-экономических или других показателей объекта автоматизации, которые должны быть достигнуты в результате создания АС, и указывают критерии оценки достижения целей создания системы.</p> <p>ГОСТ 34.601-90 АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ. СТАДИИ СОЗДАНИЯ.</p> <p>2. На этапе 1.2. "Формирование требований пользователя к АС" проводят:</p> <p>а) подготовку исходных данных для формирования требований АС (характеристика объекта автоматизации, описание требований к системе, ограничения допустимых затрат на разработку, ввод в действие и эксплуатацию, эффект, ожидаемый от системы, условия создания и функционирования системы);</p> <p>б) формулировку и оформление требований пользователя к АС</p> <p>3. На этапе 1.3. "Оформление отчёта о выполненной работе и заявки на разработку АС (технико-технического задания)" проводят оформление отчета о выполненных работах на данной стадии ... (по ГОСТ 7.32-2001)</p>
<p>Функциональные требования</p>	<p>2.6.2. В подразделе «Требование к функциям (задачам)», выполняемым системой, приводят:</p> <p>а) по каждой подсистеме перечень функций, задач или их комплексов (в том числе обеспечивающих взаимодействие частей системы), подлежащих автоматизации; ...</p>
<p>Внешние интерфейсы</p>	<p>2.6.1.1. В требованиях к структуре и функционированию системы приводят:</p> <p>в) требования к характеристикам взаимосвязей создаваемой системы со смежными системами, требования к ее совместимости, в том числе указания о способах обмена информацией (автоматически, пересылкой документов, по телефону и т. п.);</p>

Описание соответствий - 2

Атрибуты качества	<p><u>2.6.2. В подразделе «Требование к функциям (задам)», выполняемым системой, приводят:</u></p> <p>в) требования к качеству реализации каждой функции ;,; характеристики необходимой точности и времени выполнения :</p> <p><u>2.6.1.3. В требованиях к показателям назначения АС приводят значения параметров, характеризующие степень соответствия системы ее назначению.</u></p> <p>а) степень приспособляемости системы к изменению процессов № методов управления, к отклонениям параметров объекта управления; б) допустимые пределы модернизации и развития системы;</p>
Ограничения	<p><u>2.6.3. В подразделе «Требования к видам обеспечения» в зависимости от вида системы приводят требования к:</u></p> <ol style="list-style-type: none">1) математическому,2) информационному,3) лингвистическому,4) программному,5) техническому,6) метрологическому,7) организационному,8) методическому и другие видам обеспечения системы. <p>Другие подразделы допускают включение различных ограничений.</p>

Дополнительные заключения

- Пользовательские требования, в том виде, в котором они определены в классификации приведенной К. Вигерсом, не имеют **явного** отражения в документе Техническое задание ни в других документах ГОСТ.
- Этап работ «1.2. Формирование требований пользователя к АС» на стадии «1. Формирование требований к АС» (ГОСТ 34.601-90. АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ СТАДИИ СОЗДАНИЯ.) не является эквивалентом пользовательских требований «по Вигерсу».
- При необходимости, с определенными оговорками, в качестве контейнера для вариантов использования может быть использован документ «Описание постановки задачи» (см. РД 50-34.698-90).

Выводы и практические аспекты использования классификации требований

1. Использование «классификации Вигерса», в качестве основы, позволит **отобразить** ее практически на любые структуры документов, определяемые стандартами и методологиями.
2. Группировка требований по уровням и типам способствует улучшению качества формулировок самих требований, заставляя разработчиков требований **формировать дискретные требования** и следовать правилам, сформулированным в «Characteristics of Excellent Requirements» (см. **[Вигерс, 2003]**).
3. Группировка требований по уровням и типам, при условии дискретизации требований, **позволяет устанавливать связи трассировки требований**.
4. Группировка требований по уровням и типам **позволяет эффективно использовать инструментальные средства**, предназначенные для работы с требованиями, в т.ч. для автоматической генерации документации требований.

Выводы и практические аспекты использования классификации требований - 2

5. Классификация требований, в сочетании с их дискретным представлением позволяет **управлять именно ТРЕБОВАНИЯМИ а не документами.**
6. При «бесструктурном» представлении требований **сложно проводить анализ влияния**, при изменении требований, ввиду проблематичности установления связей трассировки.
7. При «бесструктурном» представлении требований **затруднительно обеспечить тестирование** созданного ПО, ввиду сложности трассировки требований со сценариями тестирования.
8. Следует обратить внимание на возможность использование на практике стандарта **IEEE 830** при отсутствии ограничений со стороны заказчика на формат документации требований.
9. Применение вариантов использования (use cases) **не всегда может быть оправдано** (зависит от проекта). Пользовательские требования могут быть сформулированы и по-другому.

Список литературы

1. **[Leffingwell & Widrig, 2003]** – Dean Leffingwell, Don Widrig. Managing Software Requirements: A Use Case Approach, Second Edition. Addison Wesley (ISBN 0-321-12247-X), 2003
2. **[Вигерс, 2003]** – Карл И. Вигерс. Разработка требований к программному обеспечению. Издательско-торговый дом “Русская редакция”, перевод на русский язык второй редакции книги – Microsoft Corporation (ISBN 5-7502-0240-2), 2004. Оригинальное издание на английском языке: Software Requirements. Second Edition. Karl E. Wiegers, Microsoft Press (ISBN 0-7356-1879-8), 2003.
3. **[RUP, 2003]** – Rational Unified Process, Version 2003.06.13
4. **[SWEBOOK, 2004]** – Guide to Software Engineering Body of Knowledge (SWEBOOK). IEEE Computer Society, 2004.
5. **[IEEE 830, 1998]** – IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998.
6. **[ГОСТ 34.003, 1990]** – Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Термины и определения. ГОСТ 34.003-90, Государственный Стандарт Российской Федерации, 1999. Госстандарт России, Москва, 1990.
7. **[ГОСТ 34.602, 1989]** – Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. ГОСТ 34.602-89, Государственный Стандарт Российской Федерации, 1999. Госстандарт России, Москва, 1990.